METHOD AND SYSTEM FOR PRIORITISED ~~CONGESTION CONTROL IN A SWITCHING~~ HUB

The present invention relates to a communications network capable of transferring data in accordance with a transfer priority number.

Currently, the majority of local area networks (LANs) operate using one of two types of protocol. These are "Carrier Sense Multiple Access Collision Detection" type networks (CSMA/CD), such as Ethernet, or token passing networks, such as Token Ring.

In point-to-point type CSMA/CD networks, communication between two end stations is by transferring data along a set transfer path defined by a number of network nodes. These nodes operate by receiving data from either a previous node, or an end station, determining the intended destination and transferring the data to the next node in the path.

However, the nodes are usually common to several paths and, as a result, when a large amount of data is to be transferred the nodes can become overloaded. When this occurs, the node is receiving data faster then the data can be retransmitted. Whilst some temporary storage space may be provided, this will be rapidly used up meaning that any incoming data will simply be lost.

In order to overcome this problem, the network nodes are configured to generate a stop signal when congestion occurs. As a result, all nodes which are transferring data to the congested node stop transmitting data, thereby allowing the node to clear any stored data. Once the congestion is cleared, the node transmits a start signal and the nodes resume the data transfer.

This method suffers from the main drawback that data transmission is completely halted. This can cause interruption of services and leads to delays in the overall transmission time. Furthermore data is often lost during the stop start procedure.

In contrast to this, token passing networks operate by passing a token around each end station of the network in turn. End stations are then only able to transmit data if they are in possession of the token. Accordingly, the amount of data on the network at any one time tends to be limited, thereby reducing the risk of congestion. Furthermore token passing protocols also operate a priority system in which data is assigned a priority. Accordingly, if the network becomes busy, the end stations are configured such that only higher priority data is transmitted. However, the switching nodes of such a network are passive during this procedure, and if an individual node becomes congested, data transferred to this node will be lost.

Accordingly, whilst both of the above mentioned protocols have separate system for dealing with the problems of congestion and while these work on their own level, they are currently incompatible and both suffer from certain disadvantages.

In accordance with a first aspect of the present invention, we provide a communications network for transferring data in accordance with a transfer priority number, the network having a number of switching nodes which transfer data transmitted between end stations coupled to the network, each switching node comprising:

a store for storing data prior to transfer;

a monitor for monitoring the volume of data being transferred through the switching node;

a comparator for comparing the volume of data to a predetermined threshold; and,

a signal generator for generating a congestion signal if the respective volume of traffic exceeds the predetermined threshold, wherein the adjacent switching nodes and/or end stations are responsive to the congestion signal to temporarily store at least some of the data to be transferred via the respective switching node, the data for storage being selected in accordance with the priority number.

In accordance with a second aspect of the present invention, we provide a method of transferring data via a communications network in accordance with a transfer priority number, the network having a number of switching nodes adapted to transfer data transmitted between end stations coupled to the network, the method comprising the steps of:

causing each switching node to monitor the volume of data being transferred therethrough;

comparing the volume of data to a predetermined threshold;

causing a switching node to generate a congestion signal if the respective volume of traffic exceeds the predetermined threshold, wherein the adjacent switching nodes and/or end stations are responsive to the congestion signal to temporarily store at least some of the data to be transferred via the respective switching node, the data for storage being selected in accordance with the priority number.

Accordingly, we provide apparatus and a method for transferring data that reduces the problem of network congestion. This is achieved by configuring the switching nodes to generate a congestion signal when the volume of data passing through the switching node becomes too great. As a result of this, adjacent switching nodes and/or end stations will reduce the amount of data to be sent by temporarily storing data which has a low transmission priority, whilst maintaining the transmission of more important high priority data.

This ensures that the transfer of important data, such as voice or video communications, or the like, can be maintained, whilst less important data is temporarily stored. Because the data is stored prior to reaching the congested node, the likelihood of losing data is vastly reduced.

Furthermore, this method can advantageously be applied to a network which transfers data in accordance with both CSMA/CD and token passing protocols. This is can be achieved using the present priority indication implemented in token passing protocols. Furthermore, a notional priority can be assigned to data defined in accordance with alternative protocol, such as CSMA/CD traffic. Alternatively, the Ethernet data packets may be given a higher or lower priority such that they are transferred either in preference to, or in deference to the Token Ring data packets.

It will also be realised that the priority numbers could be assigned to data in accordance with a number of different criteria. Thus for example, the priority could be determined on the basis of the source and/or destination address of the data. In this case virtual LANs can be defined within the overall communications network, and the priority number can therefore be defined frames originating from or addressed to end stations or other devices within these virtual LANs identified by their priority number. This would allow traffic to and from certain end stations to be given priority over traffic to and from other, less important, end stations.

Another criteria that could be used is frame size. This would allow, for example, large frame sizes to be transferred in preference to small frames.

Typically the signal generator is adapted to generate an end-of-congestion signal when the respective volume of traffic falls below a second predetermined threshold, the adjacent switching nodes being responsive to the end-of-congestion signal to transfer the temporarily stored data, the data being accessed from the store in accordance with the priority number. This ensures that any stored data is transmitted once the congestion has cleared and that furthermore, the data is accessed in such a way as to maintain the priority of the transfer as far as possible.

Typically the second threshold corresponds to a lower volume of traffic than the first predetermined threshold. By using a lower threshold for controlling the resumption of data transmission, this gives the congested node an opportunity to clear any backlog of stored data, thus ensuring that the switching node does not simply become congested again as soon as the adjacent nodes start transferring the stored data. Alternatively however, the second threshold may equal the first predetermined threshold.

Typically the monitor monitors the amount of data stored in the store. This provides an indication of how much more data the switching node will be able to store before incoming data is lost. Alternatively however, the monitor could be adapted to monitor the absolute volume of data being transferred through the switching node. This would allow the congestion signal to be generated when the switching node is unable to transfer the incoming volume of data.

Typically the predetermined threshold comprises a number of predetermined sub-thresholds, the congestion signal including an indication of the sub-threshold which has been exceeded, and wherein the data to be temporarily stored is selected based on the sub-threshold exceeded and the priority number. By providing a graded scale of thresholds, the system of the invention can begin delaying the transfer of very low priority data at the first signs of congestion arising. This helps slow or even reverse the onset of congestion before it becomes a problem, thereby ensuring that the transfer of high priority data is always maintained. Data having a progressively increasing priority is then stored if congestion continues to increase, and subsequent sub-thresholds are exceeded. This helps increase the rate at which congestion is overcome.

In accordance with a third aspect of the present invention, we provide an end station for coupling to a communications network which transfers data in accordance with a transfer priority number, the communications network being adapted to monitor the volume of data being transferred there through and to generate a congestion signal if the respective volume of traffic exceeds a predetermined threshold, the end station comprising:

a store for storing data;

an interface for coupling the end station to the communications network; and,

a processor responsive to the congestion signal to cause the end station to temporarily store at least some of the data to be transferred to the communications network, the data for storage being selected in accordance with the priority number.

Thus, the present invention further provides an end station which helps further reduce the problem of congestion on communications networks. When the network becomes congested the end station reduces the amount of data to be sent by temporarily storing data which has a low transmission priority, whilst maintaining the transmission of more important high priority data.

This ensures that the transfer of important data, such as voice or video communications, or the like, can be maintained, whilst less important data is temporarily stored. Because the data is stored prior to reaching the congested network, the likelihood of losing data is vastly reduced.

The end station is generally further adapted to respond to an end-of-congestion signal to transfer the temporarily stored data, the data being accessed from the store in accordance with the priority number.

Typically the processor generates the data to be transferred. However, the data to be transferred may generated by an alternative device, independent of the end station, or an alternative processor within the end station. Thus, for example, the data may be generated by the host processor of the end station, with an additional processor being used to control the data transfer.

It will be realised that the end station may be a computer, a LAN, or any suitable communications device.

Examples of the present invention will now be described with reference to the accompanying drawings, in which:-

Figure 1 shows in schematic form an example of a LAN operating in accordance with the present invention;

Figure 2 shows one of the switching nodes of the LAN of Figure 1 in greater detail;

Figure 3a shows an example of an Ethernet data packet suitable for transfer over the LAN of Figure 1;

Figure 3b shows an example of a Token Ring data packet suitable for transfer over the LAN of Figure 1; and,

Figure 4 shows in schematic form an example of an end station suitable for coupling to the LAN of Figure 1.

The LAN shown in Figure 1 includes a number of switching nodes 4a, 4b, 4c, 4d, 4e, 4f, 4g, 4h each having a number of ports 2. Some of the ports 2 are used to couple the switching nodes 4a, 4b, 4c, 4d, 4e, 4f, 4g, 4h together via connections 3, whilst the remainder are used for coupling end stations to the LAN. Each port 2 will include a port controller (not shown) which is used to configure the port for communication with the respective device.

As mentioned above, coupled to some of the ports 2 are a number of Ethernet end stations 5, which operate in accordance with the Ethernet protocol, a number of Token Ring end stations 6, which operate in accordance with a token passing protocol and a number of "Multigig" end stations 7, which are capable of operating in accordance with either the Ethernet or Token Ring protocols.

Each end station 5,6,7 includes a respective end station port (not shown) for coupling the end station to the network. In each case, the end station may take the form of a personal computer, a file server, a communications network or the like.

The LAN generally also includes a router 8, which is used for coordinating the transfer of data between the different end stations 5,6,7 and which is coupled to the network via one of the switching nodes 4.

When an end station 5,6,7 is initially coupled to one of the ports 2 of the switching nodes 4a, 4b, 4c, 4d, 4e, 4f, 4g, 4h, the respective port controller determines the protocol with which the end station 5,6,7 can communicate and configures the port accordingly.

The port controller also sends an indication of the nature of the respective end station 5,6,7 to the router 8. This information, is then used to coordinate the transfer of data between end stations such that two end stations will always attempt to communicate using the same protocol.

An example of a switching node 4, which is suitable for use as one of the switching nodes 4a, 4b, 4c, 4d, 4e, 4f, 4g, 4h, used in the LAN, is shown in Figure 2. This comprises a receive interface 20 and a transmit interface 21 both of which are coupled to the number of ports 2 via a bus 22. The transmit and receive interfaces are also coupled to each other by a connection 23, as well as via a buffer memory 24. A processor 25, which controls the overall operation of the switching node is also coupled to the transmit and receive interfaces 20,21 as well as the buffer memory 24.

In use, data is transferred between end stations connected to the LAN, via the switching nodes 4, in the form of data packets. In the present example the LAN is configured to transfer data packets generated in accordance with either the Ethernet or Token Ring protocols.

An example of a suitable Ethernet data packet 40 is shown in Figure 3a and this comprises a packet header 41, including a destination address field 42 a payload 43, which contains the data to be transferred, and a preamble 44.

An example of a suitable Token Ring type data packet 50 is shown in Figure 3b and this comprises a packet header

51, including a destination address field 52 and a priority number field 53 and a payload 54, which contains the data to be transferred. The priority number contained in the priority number field 53 is used to indicate the quality of service that is required by the respective data and usually consists of a number between "0" and "7". In this case 0 indicates the highest priority and this data will be transferred in preference to data having a lower priority.

It will however be realised that the present invention can be applied to any number of different protocols and should not be limited to CSMA/CD and token passing type protocols only.

In use, the switching node 4 will receive a data packet from either an end station 5,6,7 or an alternative switching node 4 at a respective port 2. The data packet is then transferred from the port 2, via the bus 22, to the receive interface 20 which acts to determine where the data packet is to be transferred to.

In general, there are two different types of destination. The first destination type is the processor 25 of the switching node itself. In this case the receive interface 20 removes the packet header and passes the payload onto the processor 25 for further processing.

Alternatively, the data packet is intended for transfer to an external destination, such as an alternative switching node 4 or an end station. In this case, the receive interface uses the destination address field 42,52 in the packet header 41,51 to determine the intended external destination of the packet.

The manner in which this is achieved will depend on the manner in which transfer of the data between end stations is coordinated.

In general, data is transferred between two end stations on a point-to-point network such as the LAN, by establishing at least one path through the network, that links the two end stations. The establishment of this path may be achieved by any one of a variety of methods, such as source routing, and will generally depend on the protocols being implemented by the network. However, once a path is determined, this will be defined in one of two ways.

Firstly, each packet generated by the transmitting end station can include route information within the packet header. This would generally take the form of a list of the switching nodes forming the path. Accordingly, when the receive interface receives such a data packet, the receive interface 20 will simply examine the header and determine therefrom the next destination on the list.

Alternatively, the path information may be stored in a memory 26 which is coupled to the receive interface, with the data packet simply including an indication of the intended destination end station 5,6,7. Accordingly, upon receipt of such a data packet, the receive interface determines the intended destination of the data packet and then uses this information to look up the next destination on the transfer path from the memory 26.

It will be realised by a person skilled in the art that the operation to determine the intended destination is not however essential to the present invention.

Once the destination is determined, the receive interface transfers the data packet to the transmit interface 21 either via the connection 23, or via the buffer memory 24 depending on the current status of the transmit interface 21. Simultaneously an indication of the intended data packet destination is transferred to the processor 25.

In order to determine the status of the transmit interface, as soon as the transmit interface is idle, a request signal is sent to the processor 25 requesting a data packet for transmission. The processor monitors the transfer of the data packets by maintaining a list of data packets stored in the buffer memory, along with an indication of their intended destination. This information is stored in a processor memory 26, which is coupled to the processor 25.

Accordingly, if there are any data packets currently stored in the buffer memory 24, the processor causes a data packet to be read out of the buffer memory 24 whilst simultaneous transferring an indication of the intended destination of the data packet to the transmit interface 21. The list of stored data packets is also updated.

Alternatively, however if the transmit interface is idle and no data packets are stored in the buffer memory, the processor 25 will transfer the request signal to the receive interface 20. Accordingly, the receive interface simply transfers the received data packet directly to the transmit interface. The intended destination of the data packet is also passed directly to the transmit interface.

In order to achieve the flow control of the present invention, the processor 25 monitors the volume of data being transferred by the switching node 4. This may be achieved in one of two ways.

Firstly, as mentioned above, the processor 25 may simply determine the volume of data from the list of data packets stored in the buffer memory 24. Alternatively, the processor 25 may monitor the volume of incoming data, or even the ratio of the number of data packets being received compared to the number of packets being transmitted during a set period of time.

In any event, either method allows the processor 25 to determine the current volume of traffic being transferred by the switching node 4. The processor 25 then compares this indication of the current volume of traffic to a predetermined threshold level.

The predetermined threshold level is an indication of substantially the maximum volume of traffic that the switching node 4 is successfully able to transfer. Thus, it may represent the maximum (or at least near maximum) amount of data that can be stored in the buffer memory 24, or alternatively the maximum volume of incoming traffic that the receive interface 20 is able to handle.

If the volume of traffic exceeds this predetermined threshold, then this indicates to the processor 25 that the switching node 4 is reaching an overload state in which it will be unable to successfully transfer all the data received at the ports 2.

Accordingly, the processor 25 generates a congestion data packet indicating that congestion is occurring within the node. This is then transferred immediately to the transmit interface 21 with an indication of the destinations to which the packet is to be transferred. The transmit interface then adds on a suitable header and transfers the signal to the adjacent switching nodes.

Thus, in the example of Figure 1, if the switching node 4a became congested, the congestion packet would be transferred to the adjacent switching nodes 4b,4c which are coupled to the switching node 4a.

The adjacent switching nodes 4b,4c, which are identical to the switching node 4a, receive the congestion packet and transfer it to their respective processors via the receive interface. The congestion packet indicates to the switching nodes 4b,4c that the switching node 4a is congested and that therefore the amount of data to be transferred to the switching node 4a should be reduced.

Accordingly, the switching nodes 4b,4c will store any data having a priority below a predetermined level in their respective buffer memories. Transfer of the higher priority data is maintained. As a result the amount of data received by the switching node 4a is reduced allowing the switching node 4a to clear any backlog of data stored in its buffer memory 24.

Throughout this procedure, the processor 25 of the switching node 4a will continue to monitor the volume of traffic being transferred through the switching node. A second threshold is defined which represents the volume of data that the switching node is able to successfully handle following a congestion problem.

Accordingly, when the volume of data passes below this second threshold, the processor determines that the congestion has been overcome and accordingly generates a clear data packet. This is transferred via the transmit interface 21, the bus 22 and the respective ports 2 to the switching nodes 4b,4c.

It should be noted that the second threshold generally corresponds to a lower volume of data than the first threshold. This ensures that the switching node 4a is able to transfer some of the data packets which are stored in the buffer memory 24 before full transfer of data to the switching node 4a is resumed. This prevents the switching node 4a simply returning to a congested state when the adjacent switching nodes 4b,4c return to transferring all the data to be transferred.

The clear packet which is received by switching nodes 4b,4c is passed to the respective processors which determine that the congestion has been cleared. Accordingly, the switching nodes 4b,4c resume sending all the data packets to the switching node 4a, as required.

It will be realised that this involves accessing the data packets stored in the buffer memory 24. In order to attempt to maintain the quality of service required by the respective priority levels of the data, the transfer of data packets from the respective buffer memories 24 of the switching nodes 4b,4c, is carried out in accordance with the priority number of the data packets.

It will also be noted that only the Token Ring type data packets include a priority number. Accordingly, the Ethernet type data packets are assigned a notional mid-level priority, which, for example, will be the number "4". The processors 25 of the switching nodes 4 are therefore configured to handle any Ethernet data packets as though they were Token Ring data packets having a priority number "4".

Alternatively, the Ethernet data packets may be given a higher or lower priority such that they are transferred either in preference to, or in deference to the Token Ring data packets.

Whilst the switching node 4a is in a congested state, the adjacent switching nodes 4b,4c will only transmit data to the switching node 4a that has a priority above a predetermined level. Accordingly, the amount of data the adjacent end stations are able to transmit is limited. This means that the adjacent switching nodes 4b,4c are more likely to enter a congested state themselves.

Thus for example, switching node 4b may enter a congested state, in which case, it will generate a congestion packet which is transferred to the switching nodes 4a,4d. Similarly, switching node 4c may become congested and generate a congestion packet which is transferred to the switching nodes 4a,4d,4e.

As a result of this, in a worst case scenario, the congestion condition will propagate through the LAN. This is in fact desirable as it allows the original source of the congestion to clear more quickly than would otherwise be the case. With the original congestion source returning to a non-congested state, the remainder of the LAN soon also returns to an uncongested state. Furthermore, the back propagation helps reduce the loss of any data that occurs when uncontrolled congestion arises in the network. This occurs whilst the flow of the high priority data is maintained.

It will be realised from this that the end stations can be advantageously adapted to respond to congestion signals. In this case, a network interface card, which is used to link the end station to the LAN via the end station port, could be adapted to prevent the transmission of low priority data onto the network when a congestion packet is received. An example of an end station 5,6,7 including a suitable network interface card is shown in Figure 4.

In this example the end station comprises a receive interface 60 and a transmit interface 61 both of which are coupled to one of the switching nodes 4 via a bus 62 and an end station port 63. The transmit and receive interfaces are also coupled to a host processor 64, via a bus 65 and respective buffer memories 66,67. A connection 68 is provided between the receive and transmit interfaces.

In use, the end station 5,6,7 will receive a data packet from one of the switching nodes 4 of the LAN, via the end station port 63. The data will be transferred to the receive interface which acts to determine the intended destination of the data packet.

If the end station is not the intended destination of the data packet, the data packet is simply transferred to the transmit interface via the connection 68 and retransmitted onto the LAN. If the data is intended for the end station, the receive interface removes the header from the data and transfers the packet payload to the host processor 64 for further processing. The buffer memory 66 is provided for temporary storage of the data packet payloads in the event that the host processor 64 is unable to receive the data for immediate processing.

Transmission is achieved by having the processor generate the data payload and transfer this to the transmit interface 61. The transmit interface adds on the suitable header and transfers the data to the LAN. Again the buffer memory 67 is provided for temporary storage of the data in the event that the transmit interface is unable to transmit the data immediately.

In the case where the switching node 4 to which the end station 5,6,7 is connected generates a congestion data packet, this will be detected by the host processor 64. It will be realised that as the different end stations 5,6,7 operate in accordance with different protocols, it is necessary for the processor 25 and the transmit interface 21, of the switching node 4, to generate a congestion packet in accordance with the correct protocol. In the case where multiple end stations are coupled to a single switching node 4, this may require the switching node to generate several different congestion packets in accordance with different protocols.

The host processor 64 responds to a congestion packet by storing any data having a priority below a predetermined level in the buffer memory 67. In the unlikely event that the amount of data in the buffer memory 67 exceeds a predetermined level, the processor 64 will halt processing of the low priority data to ensure that no data is lost.

Once the clear data packet (which must also be in a format suitable for receipt by the end station 5,6,7) is generated by the switching node 4, the host processor 64 causes the low priority data to be transferred from the buffer memory 67 to the transmit interface. Processing of the low priority data by the host processor 64 can then be resumed.

This would help further reduce the loss of any data, as well as improving the recovery time of the network by reducing the overall amount of data flowing therethrough.

In a further development of the present invention, instead of using a single threshold, the processors 25, of the switching nodes 4, would compare the volume of data currently being transferred to a number of sub-thresholds, with each sub-threshold corresponding to a different volume of data. In this case, when the lowest sub-threshold is exceeded, the processor 25 of the respective switching node 4 would generate a first congestion signal which causes adjacent nodes to stop transferring the lowest priority data only.

Similarly, if the next sub-threshold is exceeded a further congestion packet is generated causing the transmission of data packets having the next level of priority to be halted, and so on. In this manner a sub-threshold is provided corresponding to each level of priority. This allows the amount of data for which transfer is halted to be progressively increased to thereby ensure that the disruption to the transmission of high priority data will be minimised as far as possible.

In such a case, a corresponding number of second sub-thresholds will also be provided, with each second sub-threshold corresponding to level below which the volume of data being transferred must fall, in order to allow data transfer of a respective priority to be resumed.